UPM ETSIInf. Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software.

Examen Parcial de Programación II – Ejercicio Práctico

18 de Mayo de 2016

Duración: La duración total del ejercicio será de **1 hora y 30 minutos**.
Calificaciones: Las notas se publicarán el día 24 de Mayo.
Revisión: La revisión tendrá lugar el día 30 de Mayo a las 11:00 en el aula que se indicará en su momento**.**

- **Hoja de respuestas:** El código solicitado debe escribirse en los espacios señalados en la **hoja de respuestas**. Pueden usarse hojas en blanco adicionales como borrador. Sólo hay que entregar la hoja de respuestas.

Ejercicio Práctico - SOLUCIÓN

Se necesita desarrollar una aplicación de juegos de azar de tipo *Bonoloto.* En estos juegos se trata de hacer una *apuesta* que consiste en la elección de una *combinación* de números en un rango de valores posibles con la finalidad de acertar una combinación de números llamada *combinación ganadora*. Según se consiga acertar más o menos números, así será el premio ganado. Se define una *Loto* como un juego de azar que consiste en acertar *n* números diferentes en el rango [*min, max*]. A partir de esta definición general podemos definir juegos de azar más específicos particularizando una *Loto* y añadiendo más atributos y métodos. La definimos en Java así:

```
public class Loto
{
  private int[] numeros;
  private int min, max;   // Rango de los números
  public Loto (int n, int min, int max)
  {
    this.min = min;
    this.max = max;
    numeros = new int[n];
  }

 // POST: resultado es el limite inferior del rango de los números.
  public int getMin ()
  ...
 // POST: resultado es el limite superior del rango de los números
  public int getMax ()
 // PRE: 0<=pos<size()
 // POST: resultado es el elemento del objeto que esta en la
          posición "pos".
   public int get (int pos)
  ...
 // POST: resultado es un string que contiene todos los atributos.
  public String toString ()
  ...
 // POST: Decide si "x" forma parte de la Loto
  public boolean aparece (int x)
  ...
}
```

En los juegos de azar existentes actualmente, como la Lotería Primitiva, la Bonoloto o el EuroJackpot, además de los n números hay que acertar uno o más números *Extra* que tienen diferentes significados, rangos de valores y nombres según el juego de que se trate: *complementario*, *sol* o *reintegro*. Definimos un *Extra* como el *número* (int) y los límites *min* (int) y *max* (int) que marcan el rango en el que se mueve [*min, max*]. Se define en Java como:

```
public class Extra
{
  private int numero;
  private int min;
  private int max;

// PRE: min<=numero<=max
 public Extra (int numero, int min, int max) throws FueraDeRango
 ...
// POST: resultado es el numero del Extra.
 public int getNumero ()
 ...
}
```

Se define la *Primitiva* en este ejercicio como un juego de azar que consiste en acertar 6 números diferentes entre 1 y 49, mas dos *extras* que son el *complementario (*un número entre 1 y 49) y el *reintegro* (otro número entre 0 y 9). Definimos esta clase como subclase de la clase *Loto*:

```
public class Primitiva extends Loto
{
  private Extra complementario, reintegro;
  public Primitiva (int comple, int reinte) throws FueraDeRango
  {
    super(6,1,49);
    complementario = new Extra(comple,1,49);
    reintegro      = new Extra(reinte,0,9);
  }
// POST: Proporciona el complementario de la Primitiva
 public Extra getComplementario ()
  ...
 }
```

Se define el *EuroJackpot* en este ejercicio como un juego de azar que consiste en acertar 5 números diferentes entre 1 y 50, más dos *Extras* llamados *soles* que son dos números entre 1 y 10. Definimos esta clase también como subclase de la clase *Loto*:

```
public class EuroJackPot extends Loto
{
  private Extra sol1, sol2;
  public EuroJackPot (int s1, int s2)  ...
  ...
}
```

UPM ETSIInf. Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software.

Examen Parcial de Programación II – Ejercicio Práctico

18 de Mayo de 2016 – HOJA DE RESPUESTAS

Apellidos _____

Nombre _____

DNI/NIE/Pasaporte _____ **Nº Matrícula** _____

Ejercicio 1 (2 puntos)

Define en Java el *constructor* de la clase *Extra*. Si *número* se sale de los límites establecidos, se lanzará la excepción *FueraDeRango*.

Se supone ya definida la clase:

```
public class FueraDeRango extends Exception {
  public FueraDeRango (String msj) {
    super(msj);
  }
}

/*
 * PRE: min<=numero<=max
 */
public Extra (int numero, int min, int max) throws FueraDeRango
{
  if (numero < min || numero > max)
    throw new FueraDeRango("Numero fuera de rango");
  else
  {
    this.numero = numero;
    this.min = min;
    this.max = max;
  }
}
```

Ejercicio 2 (1,5 puntos)

Define en Java la operación *toString()* para visualizar una *Primitiva* como un *String*. Se debe utiliza como auxiliar la operación *toString()* de la clase *Loto*.

```
public String toString ()
{
  return "(" + super.toString() + "," +
    complementario.getNumero() + "," +
    reintegro.getNumero() + ")";
}
```

Ejercicio 3 (1,5 puntos)

Define en Java la operación *aparece* en la clase *EuroJackpot* que reciba un número *x* y que diga si *x* forma parte de la apuesta del *EuroJackpot* o no.

- La cabecera de la función que se pide es:

```
public boolean aparece (int x)
```

- Un número aparece en una apuesta del *EuroJackpot* si ha salido en los 5 números o en alguno de los *soles*.
- Se debe utilizar como auxiliar la función *aparece* de la clase *Loto.*

```
/*
 * POST: Decide si "x" forma parte de la EuroJackPot
 */
 public boolean aparece (int x)
 {
    return super.aparece(x) ||
           (sol1.getNumero() == x) ||
           (sol2.getNumero() == x);
 }
```

Ejercicio 4 (2 puntos)

Define en Java una operación denominada *consFrec* que reciba un array de *sorteos* de la *Primitiva* y genere un array de frecuencias con el número de veces que han aparecido todos los números de los *sorteos*. Los números a tener en cuenta para el cómputo son los 6 números del array *numeros* más el *complementario*.

- La cabecera de la función que se pide es: `public int[] consFrec (Primitiva[] sorteos)`
    - Ejemplo: Entrada (5 sorteos de *Primitiva)*:

N1	N2	N3	N4	N5	N6	COMPLEMENTARIO	REINTEGRO
1	12	20	22	24	40	6	4
4	20	31	33	40	42	22	7
6	12	25	33	42	49	31	5
20	25	30	38	42	44	6	0
22	24	31	42	45	49	33	3

Salida (array de frecuencias):

INDICE	FRECUENCIA	INDICE	FRECUENCIA	INDICE	FRECUENCIA
0	0	20	3	40	2
1	1	21	0	41	0
2	0	22	3	42	4
3	0	23	0	43	0
4	1	24	2	44	1
5	0	25	2	45	0
6	3	26	0	46	0
7	0	27	0	47	0
8	0	28	0	48	0
9	0	29	0	49	1

10	0	30	1		
11	0	31	3		
12	1	32	0		
13	0	33	3		
14	0	34	0		
15	0	35	0		
16	0	36	0		
17	0	37	0		
18	0	38	1		
19	0	39	0		

```
public int[] consFrec (Primitiva[] sorteos)
{
  int tamaño = sorteos[0].getMax() - sorteos[0].getMin() + 2;
  int[] resultado = new int[tamaño];
  for (int i = 0; i < sorteos.length; i++)
  {
    int comple = sorteos[i].getComplementario().getNumero();
    resultado[comple] = resultado[comple] + 1;
    for (int j = 0; j < sorteos[i].getNumeros().length; j++)
      resultado[sorteos[i].get(j)]=resultado[sorteos[i].get(j)] + 1;
  }
  return resultado;
}
```

Ejercicio 5 (3 puntos)

Define en Java la operación *masFrecuente* que reciba un array de *sorteos* de la *Primitiva* y que determine qué número ha sido el que más veces ha salido en todos ellos.

- La cabecera de la función que se pide es:

```
public int masFrecuente (Primitiva[] sorteos)
```

- Para resolver este problema, se recomienda utilizar como función auxiliar la del ejercicio 4 (*consFrec).*
- Ejemplo: Para los datos del ejemplo anterior, el resultado sería 42.

```
public int masFrecuente (Primitiva[] sorteos)
{
  int[] frecuencias = consFrec(sorteos);
  int mayorFrecuencia = 0;
  int resultado = 0;
  for (int i = 0; i < frecuencias.length; i++)
    if (frecuencias[i] > mayorFrecuencia)
    {
      mayorFrecuencia = frecuencias[i];
      resultado = i;
    }
  return resultado;
}
```